What is claimed is:

1. A processor comprising:

a functional unit adapted to execute an instruction issued to it from a dispatch stage; and

a buffer in the dispatch stage coupled to the functional unit adapted to store a plurality of the instructions before issue to the functional unit.

2. A processor according to claim 1, further comprising:

a decode stage register coupled between the dispatch stage and the functional unit, the functional unit coupled to the decode stage register for executing the instruction issued to the decode stage register from the dispatch stage.

3. A processor according to claim 1, further comprising:

control logic coupled to the buffer adapted to cause a certain one of the stored plurality of instructions to be issued to the functional unit in accordance with a loop iteration stage.

4. A processor according to claim 3, wherein the control logic is further adapted to cause the certain one of the instructions to be issued in accordance with a cycle within the loop iteration stage.

5. A processor according to claim 1, wherein the buffer is further adapted to store a plurality of loop stage bit masks respectively associated with the plurality of instructions.

6. A processor according to claim 3,

wherein the buffer is further adapted to store a plurality of loop stage bit masks respectively associated with the plurality of instructions, and

wherein the control logic is further adapted to cause the certain one of the instructions to be issued in accordance with the respective one of the loop stage bit masks associated with the certain one of the instructions.

7. A processor according to claim 4,

wherein the buffer is further adapted to store a plurality of loop stage bit masks respectively associated with the plurality of instructions, and

wherein the control logic is further adapted to cause the certain one of the instructions to be issued in accordance with the respective one of the loop stage bit masks associated with the certain one of the instructions.

8. A processor according to claim 1, further comprising:

control logic coupled to the buffer, the control logic including:

an iteration initiation register for storing a loop iteration initiation parameter;

a loop iteration register for storing a loop iteration parameter; and

a loop cycles register for storing a loop cycles parameter,

wherein the control logic is adapted to cause the plurality of instructions to be issued to the functional unit from the buffer in accordance with the loop iteration initiation parameter, the loop iteration parameter and the loop cycles parameter.

9. A processor according to claim 3, wherein the stored plurality of instructions comprises a kernel set of loop instructions, the control logic being operative so that the functional unit executes a number of loop iterations of the kernel set of loop instructions, a prologue set of loop instructions different than the kernel set of loop instructions, and an epilogue set of loop instructions different than the kernel set of loop instructions in accordance with the kernel set of loop instructions and received loop parameters.

10. A processor according to claim 9, wherein the received loop parameters include a loop iteration initiation parameter, a loop iteration parameter and a loop cycles parameter, the control logic including:

an iteration initiation register for storing the loop iteration initiation parameter;

a loop iteration register for storing the loop iteration parameter; and

a loop cycles register for storing the loop cycles parameter.

11. A processor according to claim 1, wherein the stored plurality of instructions comprises a kernel set of loop instructions, the processor further comprising:

control logic coupled to the buffer, the control logic being operative so that the functional unit executes a number of loop iterations of the kernel set of loop instructions, a prologue set of loop instructions different than the kernel set of loop instructions, and an epilogue set of loop instructions different than the kernel set of loop instructions based on the kernel set of loop instructions and received loop parameters.

12. A processor according to claim 11, wherein the received loop parameters include a loop iteration initiation parameter, a loop iteration parameter and a loop cycles parameter, the control logic including:

an iteration initiation register for storing the loop iteration initiation parameter;

a loop iteration register for storing the loop iteration parameter; and

a loop cycles register for storing the loop cycles parameter.

13. A processor according to claim 2, wherein the stored plurality of instructions comprises a kernel set of loop instructions, the processor further comprising:

control logic coupled to the buffer, the control logic being operative so that the functional unit executes a number of loop iterations of the kernel set of loop instructions, a prologue set of loop instructions different than the kernel set of loop instructions, and an epilogue set of loop instructions different than the kernel set of loop instructions based on the kernel set of loop instructions and received loop parameters.

14. A processor according to claim 13, wherein the received loop parameters include a loop iteration initiation parameter, a loop iteration parameter and a loop cycles parameter, the control logic including:

an iteration initiation register for storing the loop iteration initiation parameter;

a loop iteration register for storing the loop iteration parameter; and

a loop cycles register for storing the loop cycles parameter.

15. A processor according to claim 3, further comprising a fetch unit, wherein the control logic is further operative so that the fetch unit can be shut down during execution of the stored plurality of instructions.

16. A processor according to claim 8, further comprising a fetch unit, wherein the control logic is further operative so that the fetch unit can be shut down during execution of the stored plurality of instructions.

17. A processor according to claim 11, further comprising a fetch unit, wherein the control logic is further operative so that the fetch unit can be shut down during execution of the prologue, kernel and epilogue sets of loop instructions.

18. A processor according to claim 13, further comprising a fetch unit, wherein the control logic is further operative so that the fetch unit can be shut down during execution of the prologue, kernel and epilogue sets of loop instructions.

19. A processor according to claim 9, wherein the kernel set of loop instructions comprise MVE code.

20. A processor according to claim 11, wherein the kernel set of loop instructions comprise MVE code.

21. A processor according to claim 13, wherein the kernel set of loop instructions comprise MVE code.

22. A processor according to claim 3, wherein the control logic is further operative to allow interrupts to be handled at the end of a current loop iteration.

23. A processor according to claim 8, wherein the control logic is further operative to allow interrupts to be handled at the end of a current loop iteration.

24. A processor according to claim 11, wherein the control logic is further operative to allow interrupts to be handled at the end of a current loop iteration.

25. A processor according to claim 13, wherein the control logic is operative to allow interrupts to be handled at the end of a current loop iteration.

26. A buffer in the dispatch stage of a processor, the buffer being associated with a functional unit that executes instructions issued to it from the dispatch stage, the buffer comprising:

a first portion for storing a kernel set of loop instructions;

a second portion for storing a plurality of modulo schedule stage identifiers respectively associated with the kernel set of loop instructions.

27. A buffer according to claim 26, wherein the processor further includes control logic, the buffer being coupled to the control logic and the functional unit for causing the kernel set of loop instructions to be issued to the functional unit in accordance with the modulo schedule stage identifiers.

28. A buffer according to claim 27, wherein the modulo schedule stage identifiers comprise bit fields, the control logic determining whether to issue a certain one of the loop instructions to the functional unit in accordance with a bit position of a set bit in the bit field corresponding to the certain loop instruction.

29. A buffer according to claim 27, wherein the control logic is operative to cause a number of loop iterations of the kernel set of loop instructions, a prologue set of loop instructions different than the kernel set of loop instructions, and an epilogue set of loop instructions different than the kernel set of loop instructions in accordance with the stored modulo schedule stage identifiers.

30. A buffer according to claim 26, wherein the loop instructions comprise undecoded instructions, and wherein the processor further includes a decode stage interposed between the functional unit and the buffer for decoding the instructions.

31. A buffer according to claim 26, wherein the loop instructions comprise decoded instructions in the form of functional unit control signals.

32. A processor for executing a number of iterations of a loop, the loop including a prologue set of loop instructions, a kernel set of loop instructions and an epilogue set of loop instructions, the processor comprising:

a plurality of functional units; and

a dispatch stage coupled to the functional units for issuing instructions to the functional units, the dispatch stage including:

a plurality of buffers adapted to store the kernel set of loop instructions; and

control logic coupled to the plurality of buffers for causing the stored kernel set of instructions to be selectively issued to the functional units, the control logic being operative so that the functional units execute the number of iterations of the kernel set of loop instructions, the prologue set of loop instructions and the epilogue set of loop instructions based on the stored kernel set of loop instructions.

33. A processor according to claim 32, further comprising a fetch unit, wherein the control logic is further operative so that the fetch unit can be shut down during execution of the number of iterations of the loop.

34. A processor according to claim 32, wherein the control logic is further operative to allow interrupts to be handled at the end of a current one of the number of loop iterations, and to complete the number of loop iterations after the interrupt is handled.

35. A method for executing a number of iterations of a loop in a processor, the loop including a prologue set of loop instructions, a kernel set of loop instructions and an epilogue set of loop instructions, the method comprising:

storing the kernel set of loop instructions at a dispatch stage of the processor;

storing loop parameters in control logic associated with the stored loop instructions; and

causing the stored kernel set of instructions to be selectively issued to functional units of the processor in accordance with the stored loop parameters so that the functional units of the processor execute the number of iterations of the kernel set of loop instructions, the prologue set of loop instructions and the epilogue set of loop instructions based on the stored loop instructions.

36. A method according to claim 35, wherein the step of storing the loop parameters includes:

storing an iteration initiation parameter in an iteration initiation register in the control logic;

storing a loop iteration parameter in a loop iteration register in the control logic; and

storing a loop cycles parameter in a loop cycles register in the control logic.

37. A method according to claim 36, wherein the step of causing the stored kernel set of loop instructions to be selectively issued includes:

adjusting the value in the loop cycles register from the stored loop cycles parameter in accordance with a processor cycle; and

resetting the value in the loop cycles register in accordance with the adjusted value and the stored iteration initiation parameter.

38. A method according to claim 39, wherein the step of causing the stored kernel set of loop instructions to be selectively issued includes:

adjusting the value in the loop iteration register in accordance with the resetting step; and

completing the issue of loop instructions in accordance with the adjusted value in the loop iteration register.

39. A method according to claim 35, further comprising:

storing a set of modulo schedule stage identifiers respectively associated with the kernel set of loop instructions, the step of causing the stored kernel set of loop instructions to be selectively issued including the step of comparing the stored loop parameters with the modulo schedule stage identifiers.

40. A method according to claim 39, wherein the step of comparing the stored loop parameters with the modulo schedule stage identifiers includes the step of indexing to a certain one of the modulo schedule stage identifiers in accordance with a current cycle in the modulo schedule stage indicated by the stored loop parameters, the step of causing the stored kernel set of loop instructions to be selectively issued further including issuing the associated loop instruction to the functional unit if the certain modulo schedule stage identifier indicates that the functional unit is active.

41. A method according to claim 35, wherein the processor further comprises a fetch unit, the method further comprising:

shutting down the fetch unit during execution of the number of iterations of the loop.

42. A method according to claim 35,

allowing interrupts to be handled at the end of a current one of the number of loop iterations; and

completing the number of loop iterations after the interrupt is handled.

43. A processor for executing a number of iterations of a loop, the loop including a prologue set of loop instructions, a kernel set of loop instructions and an epilogue set of loop instructions, the processor comprising:

means for storing the kernel set of loop instructions at a dispatch stage of the processor;

means for storing loop parameters in control logic associated with the stored loop instructions; and

means for causing the stored kernel set of instructions to be selectively issued to functional units of the processor in accordance with the stored loop parameters so that the functional units of the processor execute the number of iterations of the kernel set of loop instructions, the prologue set of loop instructions and the epilogue set of loop instructions based on the stored loop instructions.

44. A processor according to claim 43, wherein the means for storing the loop parameters includes:

means for storing an iteration initiation parameter in an iteration initiation register in the control logic;

means for storing a loop iteration parameter in a loop iteration register in the control logic; and

means for storing a loop cycles parameter in a loop cycles register in the control logic.

45. A processor according to claim 44, wherein the means for causing the stored kernel set of loop instructions to be selectively issued includes:

means for adjusting the value in the loop cycles register from the stored loop cycles parameter in accordance with a processor cycle; and

means for resetting the value in the loop cycles register in accordance with the adjusted value and the stored iteration initiation parameter.

46. A processor according to claim 45, wherein the means for causing the stored kernel set of loop instructions to be selectively issued includes:

means for adjusting the value in the loop iteration register in accordance with the operation of the resetting means; and

means for completing the issue of loop instructions in accordance with the adjusted value in the loop iteration register.

47. A processor according to claim 43, further comprising:

means for storing a set of modulo schedule stage identifiers respectively associated with the kernel set of loop instructions, the means for causing the stored kernel set of loop instructions

to be selectively issued including means for comparing the stored loop parameters with the modulo schedule stage identifiers.

48. A processor according to claim 47, wherein the means for comparing the stored loop parameters with the modulo schedule stage identifiers includes means for indexing to a certain one of the modulo schedule stage identifiers in accordance with a current cycle in the modulo schedule stage indicated by the stored loop parameters, the means for causing the stored kernel set of loop instructions to be selectively issued further including means for issuing the associated loop instruction to the functional unit if the certain modulo schedule stage identifier indicates that the functional unit is active.

49. A processor according to claim 43, wherein the processor further comprises a fetch unit, the processor further comprising:

means for shutting down the fetch unit during execution of the number of iterations of the loop.

50. A processor according to claim 43,

means for allowing interrupts to be handled at the end of a current one of the number of loop iterations; and

means for completing the number of loop iterations after the interrupt is handled.